

⑫ 公開特許公報(A)

平3-33955

⑤Int.Cl.⁵

識別記号

庁内整理番号

④公開 平成3年(1991)2月14日

G 06 F 12/12

A

7232-5B

審査請求 未請求 請求項の数 1 (全9頁)

⑥発明の名称 キャッシュメモリコントローラ

⑪特 願 平1-169480

⑫出 願 平1(1989)6月29日

⑬発明者 小池 庸夫 東京都港区芝5丁目33番1号 日本電気株式会社内

⑭出願人 日本電気株式会社 東京都港区芝5丁目7番1号

⑮代理人 弁理士 内原 晋

明 細 書

1. 発明の名称

キャッシュメモリコントローラ

2. 特許請求の範囲

中央処理装置がアクセスと予測されるメインメモリの内容を、高速動作可能なキャッシュメモリへ転送しておき、このキャッシュメモリから前記中央処理装置へデータを供給することによって前記中央処理装置と前記メインメモリの動作スピードのギャップを埋めるキャッシュメモリコントローラにおいて、キャッシュデータに対し外部から入力される優先順位指定情報を記憶しておく優先順位記憶手段と、該優先順位指定情報を該優先順位記憶手段に記憶させるための制御を行い、破棄するキャッシュメモリを選択・決定する制御回路と、この制御回路に該優先順位記憶手段に記憶されている優先順位情報を参照する機能とを具備することを特徴とするキャッシュメモリコントローラ。

3. 発明の詳細な説明

(産業上の利用分野)

本発明はキャッシュメモリコントローラに関し、特にキャッシュメモリに登録されている情報の追いつ出し方を改良したキャッシュメモリコントローラに関する。

(従来の技術)

第5図^(a)にキャッシュメモリを搭載した情報処理システムの構成例を、第5図^(b)にキャッシュメモリを搭載しない情報処理システムの構成例を示す。1はキャッシュメモリコントローラ、13はキャッシュメモリのリード信号、14はキャッシュメモリライト信号を張している。25は中央処理装置(以下CPUと称す)を、26はメインメモリを、27はキャッシュメモリを、23はメインメモリへのリード信号、24はメインメモリへのライト信号、29はデータバスを張している。キャッシュメモリコントローラの動作を第6図のキャッシュメモリコントローラ制御手段の図を用いて説明する。キャッシュメモリコントローラの制御には何種類も

の制御方式があるが、ここではオンデマンド方式と呼ばれる方法について説明する。オンデマンド方式とは、CPUからアクセスがあった場合にのみキャッシュメモリコントローラが動作を開始するものである。まずステップ100でCPUからのアクセスがあるかどうかをサンプルし、アクセスがなければループする。アクセスがあった場合、ステップ101でキャッシュメモリへ登録された情報へのアクセスかどうかをチェックする。登録されている情報へのアクセスであることをキャッシュヒットと呼び、登録されていない情報へのアクセスであることをキャッシュミスヒットと呼ぶ。キャッシュヒットの場合ステップ105へ進み、キャッシュメモリのデータをCPUへ引き渡すことになる。ミスヒットした場合ステップ102へ進み、キャッシュメモリへ登録できる余地があるかどうかをチェックする。キャッシュメモリコントローラ内部では、タグメモリと称するメモリ(後述)を持っているので、このメモリに書き込みが行われていない部分があるかどうかをチェックすることになる。

ヒットが判定され、キャッシュメモリからのデータが2クロック目の最後(第3クロックの立ち上がり)にCPUへ供給されひとつのサイクルが完了することになる。本図では同様のキャッシュヒットサイクルが全部で4回連続していることを図示している。

第7図(b)は、キャッシュメモリコントローラが接続されていない場合の動作タイミングを示している。同図(a)と同様に、第1クロックの立ち上がりでアドレス、リード信号が出力され、メインメモリへのデータアクセスが開始される。メインメモリは3クロックのアクセスタイムを必要とするので、第3クロックの最後(第4クロックの立ち上がり)にメインメモリから供給され、ひとつのサイクルが完了することになる。本図でも同図(a)と同様に4回のサイクルが連続している状態を図示している。

第7図(c)はキャッシュメモリコントローラが接続されていてかつ、ミスヒットした場合の動作タイミングを示している。第1クロックでアドレス、

タグメモリが一杯でなければその一つを任意に選択しステップ104へ進む。一杯である場合は、ステップ103で既に登録されているキャッシュメモリの一つを選択することになる。選択したブロックに対しステップ104でメインメモリの内容をキャッシュメモリへコピーする動作(以下リブレース動作と称する)を行なうことになる。ステップ105でCPUへデータを引き渡して動作を終了する。

(b) 第5図のキャッシュメモリを搭載しないシステム構成例では第5図(a)に比べて、キャッシュメモリコントローラ1とキャッシュメモリ27がないこと、リード信号21、ライト信号22が直接メインメモリへ供給されている点が異なっている。

第7図(a)は、キャッシュメモリコントローラが接続されていてかつ、キャッシュヒットした場合の動作タイミングを示している。第1クロックの立ち上がりでアドレス、リード信号が出力され、CPUからのアクセスが外部に通知される。この情報からキャッシュメモリコントローラ内でキャッシュ

リード信号が出力され、アクセスが開始されると、キャッシュメモリコントローラ内でミスヒットが検出され、リブレース動作が開始する。リブレース動作では、メインメモリの内容をキャッシュメモリへ転送するためにメインメモリへのアクセスが発生し、そのサイクルは3クロックの時間を必要とする。最初にリブレースされるデータはCPUがアクセスしようとしたデータであるためにCPUへもそのデータが供給されることになる。(このような動作をフェッチバイパス動作と呼ぶ)リブレース動作は、4ワードを1ブロックとするために、あと3回のメインメモリアクセスが必要となるが、この部分は最初のアクセスに連続するアドレスのメモリブロックを転送することになる。その際、ニブルモードのダイナミックメモリのようなものを使用すると連続したメモリブロックを読み出すときは通常のアクセスよりも早いアクセスが可能であるため、3回のメモリ転送は最初のメモリ転送よりも高速に行うことができるように設計することができる。本図では、このようなモー

ドのメモリを使用し、連続するメモリアクセスには1クロックで動作が完了するものを使用した場合を示している。(このように連続するメモリアクセスが最初のアクセスよりも高速にメモリ転送ができるモードをバーストリブレース動作と称する。)3回のアクセスは上述したバーストリブレースを行うために1クロックでメインメモリからキャッシュメモリへの転送が完了している。第8クロックの立ち上がりでCPUからの次のアクセスが出力され、キャッシュヒットした動作が発生している。

キャッシュヒット時は、4ワードのデータをアクセスするのに、全部で8クロック(1サイクル2クロック×4)がかかっている。キャッシュを搭載しない場合は、全部で12クロック(1サイクル3クロック×4)、キャッシュミスヒットでリブレースが発生した場合、トータル13クロックかかることになる。

第4図に従来のキャッシュメモリコントローラの機能ブロック図を示す。本図は本発明を説明する

で、後述するリブレースメントアルゴリズムに従ってタグメモリ2の内容を制御するものである。比較器4の出力信号12は、CPUアドレス7とタグメモリ出力8を比較し、一致するとアクティブとなる信号である。

リブレースが発生した際にタグメモリのどの内容を破棄し、新しい情報を登録するかを決定するアルゴリズムをリブレースメントアルゴリズムと称するが、リブレースメントアルゴリズムにはFIFO(First-In First-Out)方式、LRU(Least Recently Used)方式などがあつた。FIFO方式は最も過去に入れ替えたブロックを最優先で追い出す方式、LRU方式とは、最も過去に参照されたブロックを最優先で追い出す方式である。第8図(a)、(b)にFIFO方式、LRU方式の制御手順を示した。

第8図(a)のFIFO方式では、リブレース時にステップ106でリブレースしたブロックを順序付けて記憶し、ステップ107のブロック選択時にステップ106で付けた順序を参照し、一番古く

ために必要な部分のみ記述している。1は従来のキャッシュメモリコントローラであり、タグメモリ2はキャッシュメモリに登録されているアドレス情報を記憶している。比較器4は、CPUからアクセスされたアドレス情報とタグメモリ2に登録されている情報とを比較し、キャッシュメモリへ登録されているアドレスへのアクセスかいなかを判定する。13と14はキャッシュメモリインタフェース5によって生成されるキャッシュメモリ制御信号で、キャッシュメモリリード信号とキャッシュメモリライト信号を表している。5はキャッシュメモリインタフェースで、比較器4の出力とリード/ライト信号21、22からキャッシュメモリリード信号13、キャッシュメモリライト信号14とメインメモリへのリード信号23、メインメモリへのライト信号24を生成するブロックである。リブレース制御回路20は、比較器4の出力結果に基づき、リブレースが必要な場合の制御を行うブロックである。7はCPUアドレス、8はタグメモリ出力、10はリブレース制御回路20の出力

リブレースされたブロックを選択する。第8図(b)のLRU方式では、キャッシュヒット時にステップ108でヒットしたブロックを順序付けて記憶し、ステップ109でブロック選択時に最も古くアクセスされたブロックを選択するようになっている。FIFO方式の場合、第4図に直接記載されていない、リブレースの起こった順序が記憶されるメモリがあり、リブレースが発生した際にリブレースを行なったブロックを記憶しておき、ブロックを選択する際にその情報から最も古くリブレースされたブロックを選択する。すなわち1回目のリブレース動作で記憶されたタグメモリには'1'が、2回目にリブレースされたタグメモリには'2'といった情報が記憶され、タグメモリを256ワードとすると、このリブレースの発生した順序を記憶しているメモリは8ビット×256ワード構成となる。次にリブレースが発生した場合、リブレース制御回路20内部でこのメモリの内容をすべてサーチして小さい値を持ったものを追い出すことによってFIFO方式が実現されることになる

(サーチする際には8ビットのデータを取り扱えばよい)。もちろん追い出す際に残ったすべてのリブレース順序の記憶メモリの内容は一律に減少せられる。同じようにLRU方式ではCPUからアクセスがあり、キャッシュヒットした場合に同様のメモリへ順序データが書き込まれ、ブロックを選択する際にその情報を参照し追い出すブロックを選択することになる。

タグメモリの内容と、CPUがアクセスする実際のデータが格納されているキャッシュメモリは1対1に対応しているので、追い出す(無効化する)タグメモリを選択することは、キャッシュメモリへ登録されているデータを追い出す(無効化)ことと等価である。本文書内ではタグメモリの追い出しと、キャッシュメモリデータの追い出しを同じものとして表現している。

〔発明が解決しようとする課題〕

上述したキャッシュメモリシステムでは、ソフトウェアタイマーのようなものでは、プログラムがキャッシュメモリに存在するかしなないかで、パスサ

ていた。

〔課題を解決するための手段〕

本発明は、中央処理装置がアクセスすると予測されるメインメモリの内容を、高速動作可能なキャッシュメモリへ転送しておき、このキャッシュメモリから前記中央処理装置へデータを供給することによって前記中央処理装置と前記メインメモリの動作スピードのギャップ⁷を縮めるキャッシュメモリコントローラにおいて、キャッシュデータに対し外部から入力される優先順位指定情報を記憶しておく優先順位記憶手段と、該優先順位記憶情報を該優先順位記憶手段に記憶させるための制御を行い、破棄するキャッシュメモリを選択・決定する制御回路と、この制御回路に該優先順位記憶手段に記憶されている優先順位情報を参照する機能とを具備することを特徴とするものである。

〔実施例〕

次に本発明について図面を参照して説明する。

第1図は本発明によるキャッシュメモリコントローラの第1の実施例の機能ブロック図を表している。

イクル数が約1.5倍(8クロック:13クロック)違うために精度がかなり悪いものとなるという問題があった。

これにもまして、FIFO方式やLRU方式のリブレースメントアルゴリズムを用いたキャッシュメモリコントローラでは、タグメモリに登録されている情報は均一に追い出される可能性があり、高速な動作を要求される処理、例えば割り込み処理等においてキャッシュに登録されていない場合応答スピードに問題があった。第7図からわかるとおり、キャッシュヒットの場合は8クロックでアクセスできるが、キャッシュミスヒット時は13クロックとメインメモリを直接接続した場合の12クロックよりもアクセスが遅くなる。また、高速応答性を確保するためにキャッシュメモリ(キャッシュメモリコントローラ)を2つ設置し、上記割り込みハンドラのような情報を一方のキャッシュメモリに固定的に登録しておくといった手法もあるが、これは2つのキャッシュメモリコントローラの内、どちらを選択するかを決める外部回路が必要になっ

第4図の従来のキャッシュメモリコントローラと同じ番号は同じ機能を表している。1Aはキャッシュメモリコントローラ、タグメモリ15と16はタグメモリ2と同じようにキャッシュメモリへ登録しているアドレス情報を記憶している。この内15は優先順位が高いもの、16は優先順位の低いものを登録するためのタグメモリを表している。リブレース制御回路6は、従来例におけるリブレース制御回路20と同様の機能ブロックを実現しているが、本実施例では2つのタグメモリを持っているためにそれに対応できるようになっている。

17と18はそれぞれのタグメモリへの制御信号で、従来例における信号10と同様の機能を持っている。優先順位指定情報9は'1'で優先順位が高く'0'で低いものとする。インバータゲート19は優先順位指定情報9を論理反転させている。この出力は優先順位の低いタグメモリ16の動作許可信号となっている。同様に優先順位指定情報はそのまま優先順位の高いタグメモリ15の動作許可信号となっている。

本実施例のキャッシュメモリコントローラの動作を説明する。CPUアドレス7とタグメモリ15または16の内容を比較し、一致するものがない場合、信号12がインアクティブとなりリブレース制御回路6が動作を開始する。この時、優先順位指定情報⁽¹²⁾がアクティブであれば優先順位の高いタグメモリ5が選択され、この中からリブレースアルゴリズムに従って追い出されるタグメモリが選択されることになる。また優先順位指定情報がインアクティブの場合、インバートゲート19によって優先順位の低いタグメモリ16が選択され、この中から追い出されるタグメモリが選択されることになる。追い出されたタグメモリに新しい情報が書き込まれることになる。このように優先順位の高いタグメモリと低いタグメモリを二つ並べておき、優先順位入力によってそれを選択している。先の割り込みハンドラのような高速な応答性を必要とするプログラムをアクセスする際には優先順位指定情報を'1'としておき、通常のプログラムなどは優先順位指定情報を'0'にしておく

することもできる。

第2図に本発明によるキャッシュメモリコントローラの第2の実施例を示す。1Bはキャッシュメモリコントローラであり、3は優先順位記憶用のメモリで、タグメモリ2と1対1に対応している。本実施例におけるリブレース制御回路6'は優先順位メモリ3の内容を参照しどのタグメモリを追いつくかを決定するアルゴリズムを持っている。9'は優先順位入力で、第1図の実施例では1本であったがここでは複数本あるものとしている。(n本あれば2ⁿの優先順位をつけることができる；ここでは2本とする。)本実施例におけるキャッシュメモリコントローラの動作を説明する。CPUアドレス7とタグメモリ8の内容が比較器4で比較され一致しない場合リブレース制御回路6'が動作を開始する。この時、どのタグメモリを追いつくかを決定する際に優先順位記憶メモリの出力11を通じて優先順位メモリ3の内容を参照することになる。実際には従来のキャッシュメモリコントローラで説明したように、FIFO方式であれ

とによって、割り込みハンドラは優先順位の高いタグメモリ15へ記憶され、通常のプログラムがどのようにアクセスされてもタグメモリ15から追い出されることはなくなる。これによって、割り込みハンドラはいつでもキャッシュメモリへ登録されていることになり、高速にアクセスすることが可能になる。

第3図にこのキャッシュメモリコントローラ1A^(a)を搭載したシステムの構成例を示す。第5図^(a)の従来のキャッシュメモリコントローラを搭載したシステムとの違いは、アドレスデコーダ30が追加され、その出力が優先順位指定情報に入っていることである。アドレスデコーダ30は割り込みハンドラやその他優先順位の高いプログラムの格納されたアドレスの時、出力がアクティブとなると、優先順位指定情報がアクティブとなり優先順位の高いタグメモリが選択されることになる。

このほかシステムの構成としては、CPUが割り込み処理に入っていることを示す信号を出力するものであれば、この信号を優先順位指定情報と

バリエーション順序を記憶するメモリ(第4図では8ビット×256ワード構成)の中で一番小さい値を持ったものを検索する機能がリブレース制御回路6'に組み込まれているが、そのビット数を $8+2=10$ ビットとし、10ビットのデータを検索する回路構成としておく。10ビットの内、下位8ビットはリブレース順序記憶メモリの内容を割当て下位2ビットに優先順位メモリ3の内容を割り当てておく。こうすることによって優先順位の高いタグメモリ⁽¹³⁾リブレース制御回路6'内の検索ブロックから見ると大きな値を持っていることになり、追い出されることがなくなることになる。またすべての優先順位が同一であれば、従来のキャッシュメモリコントローラと同様にFIFO方式で追い出されるタグメモリが選択されることになる。第9図にこの第2の実施例の優先順位の動作例を示す。キャッシュメモリへ登録される順序はデータ111-118の順番とし、111は優先順位'1'、112は同じく'1'、113は'2'、114は'0'、115は'3'、116は'2'、117

は'2'、118は'0'とする。本実施例によれば優先順位端子から入力された優先順位情報が記憶されているため比較器で検索される情報119の値に記述されている値となり、この内一番小さい値を持っている(003h:数字の後ろの'h'は16進数であることを表すものとする。)データ114が最初に追い出されることになる。

第1の実施例では、優先順位の低いブロックをタグメモリに登録する際には高い優先順位のタグメモリの中^からのみ追い出されるものが選択されたが、第2の実施例では優先順位の低いものから追い出されることになる。これは優先順位指定情報が固定された場合、第1の実施例では優先順位の低いあるいは低いタグメモリのみ使用されることになり、キャッシュメモリコントローラが内蔵しているタグメモリすべてを使用することができないことになるが、第2の実施例では優先順位端子を協定的に使用してもすべてのタグメモリにアクセスすることができる(すなわち従来のキャッシュメモリコントローラとまったく同じ動作をするこ

とができる)ことになる。

(発明の効果)

以上説明したように本発明は、タグメモリに優先順位を記憶する手段によって、リブレースが発生した際、追い出されにくいタグメモリを実現することによって、高速な応答性を要求される処理をキャッシュメモリに^(冗余り発生)常駐^{ではない}させることによって高速性を維持することができるという効果がある。

4. 図面の簡単な説明

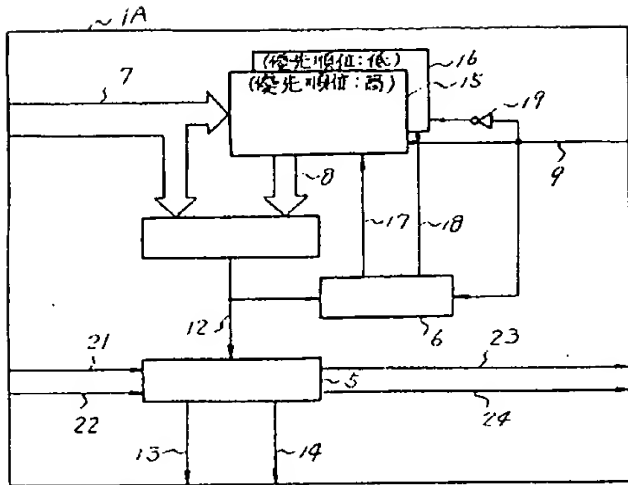
第1図は本発明の第1の実施例のブロック図、第2図は本発明の第2の実施例のブロック図、第3図は第1図または第2図のキャッシュメモリコントローラを搭載したシステムの構成例を示すブロック図、第4図は従来例のブロック図、第5図(a)は第4図のキャッシュメモリコントローラを搭載したシステムの構成例を示すブロック図、第5図(b)はキャッシュメモリコントローラを搭載しないシステムの構成例を示すブロック図、第6図は

従来例の制御手順を示すフローチャート、第7図(a), (b), (c)は従来例の動作を示すタイミングチャート、第8図(a), (b)は従来例の制御手順を示すフローチャート、第9図は第2図の実施例の優先順位の動作を示す説明図である。

1A, 1B…キャッシュメモリコントローラ、2…タグメモリ、3…優先順位メモリ、4…比較器、5…キャッシュメモリインタフェース、6, 6'…リブレース制御回路、7…CPUアドレス、8…タグメモリ出力、9, 9'…優先順位指定情報、10…リブレース制御回路出力、11…優先順位メモリ出力、12…比較器出力信号、13…キャッシュメモリリード信号、14…キャッシュメモリライト信号、15…優先順位の低いタグメモリ、16…優先順位の低いタグメモリ、17…リブレース制御回路出力、18…リブレース制御回路出力、19…インバータゲート、21…CPUからのリード信号、22…CPUからのライト信号、23…メインメモリへのリード信号、24…メインメモリへのライト信号、25…中央処理装置(CPU)、

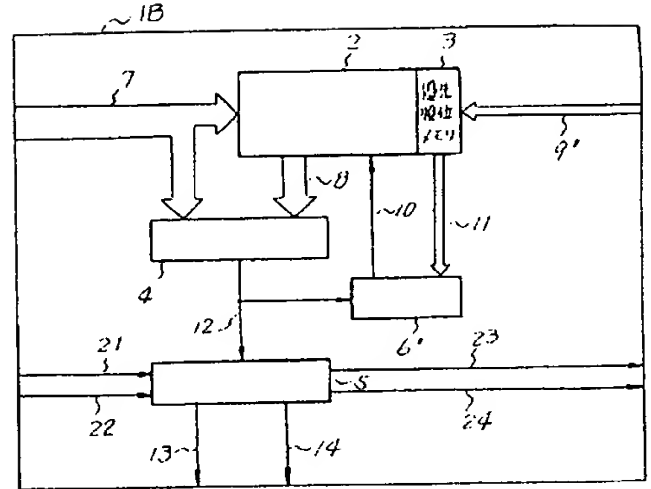
26…メインメモリ、27…キャッシュメモリ、28…アドレスバス、29…データバス、30…アドレスデコーダ。

代理人 内原 晋



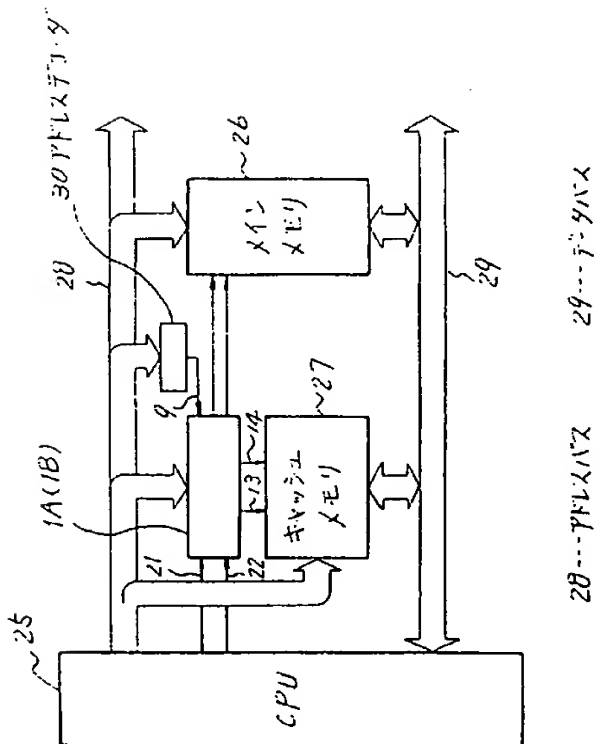
1A…キャッシュメモリコントローラ 4…比較器
5…キャッシュメモリインタフェース 6…リフレッシュ制御回路
7…CPUアドレス 12…比較器出力信号
13…キャッシュメモリリード信号 14…キャッシュメモリリード信号
21, 23…リード信号 22, 24…ライト信号

第1図

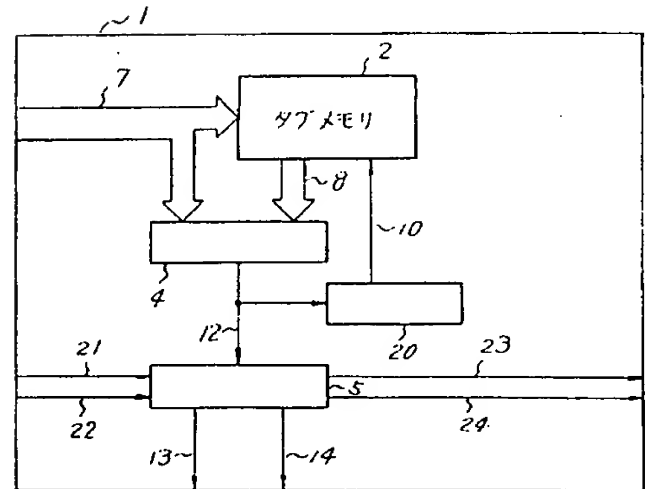


1B…キャッシュメモリコントローラ 2…タグメモリ
6…リフレッシュ制御回路 7…CPUアドレス 12…比較器出力信号
13…キャッシュメモリリード信号 14…キャッシュメモリリード信号
21, 23…リード信号 22, 24…ライト信号

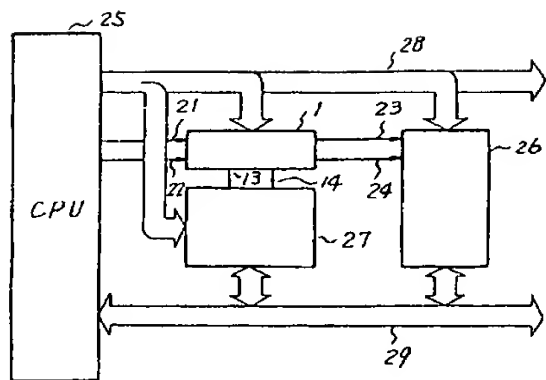
第2図



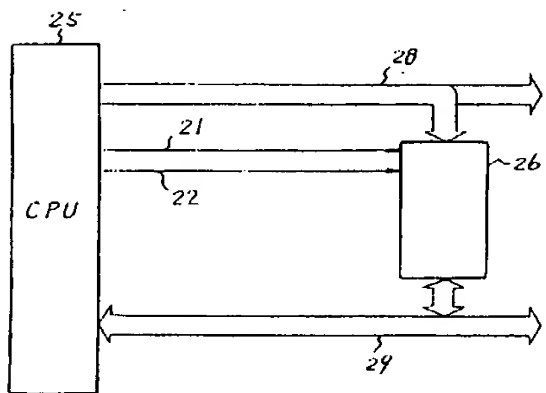
第3図



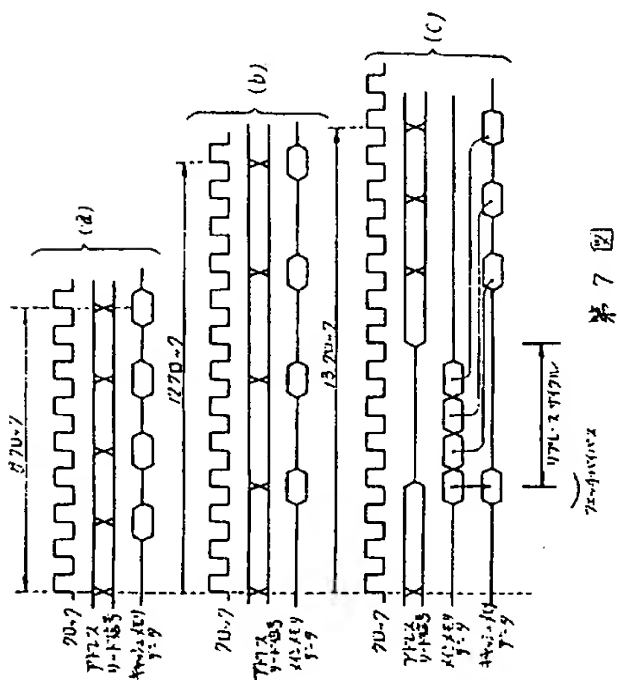
第4図



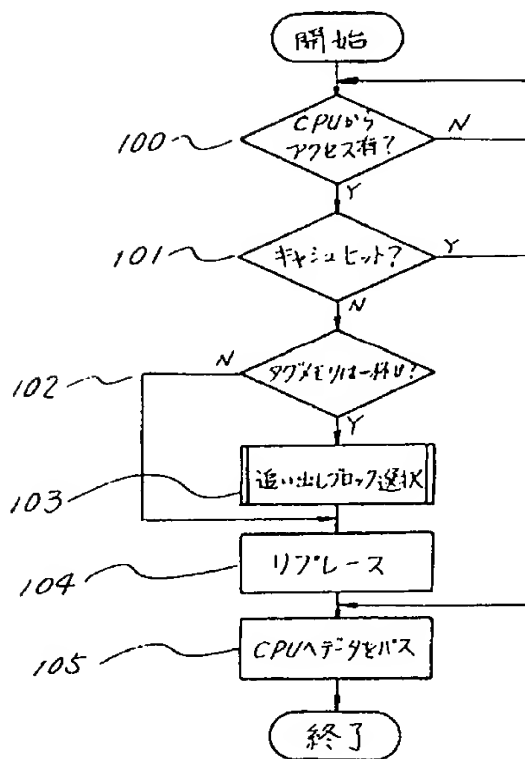
第5図(a)



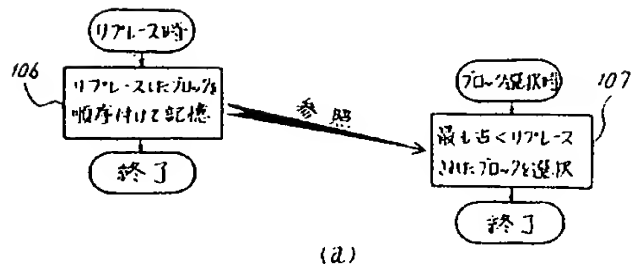
第5図(b)



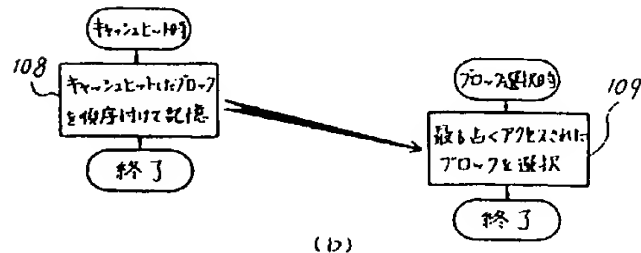
第7図



第6図



(a)



(b)

第8図

	110 設定値 出力ビット	121 111000010 出力ビット		119 比較器の出力値	
111	01	000000000	=	100h	④
112	01	000000001	=	101h	⑤
113	10	000000010	=	202h	⑥
114	00	000000011	=	003h	⑦
115	11	000000100	=	304h	⑧
116	00	000000101	=	005h	②
117	10	000000110	=	206h	⑦
118	00	000000111	=	007h	③

③ 120
設定値出力ビット

第9図